UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/708,159	11/08/2000	Toshiaki Yasue	JP919990097US1	1032

7590 03/25/2009

William A Kinnaman Jr
IBM Corporation - MS P386
2455 South Road
Poughkeepsie, NY 12601

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
03/25/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte TOSHIAKI YASUE, KAZUNORI OGATA,
KAZUAKI ISHIZAKI, and HIDEAKI KOMATSU

Appeal 2008-2940
Application 09/708,159[1]
Technology Center 2100

Decided:[2] March 25, 2009

Before JOHN C. MARTIN, JEAN R. HOMERE, and
CAROLYN D. THOMAS, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL

---

[1] Filed on November 08, 2000. The real party in interest is International Business Machines, Corp.

[2] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Date (electronic delivery).

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1 through 10. We have jurisdiction under 35 U.S.C. § 6(b). We reverse.

Appellants' Invention

Appellants invented a program execution method for transferring an execution program including a plurality of transfer points from an interpreter process to a compiled code loop process to thereby optimize the loop process by changing it from an irreducible state to a reducible state. (Spec. 5: 1-4; App. Br. 6: 1-7.) Particularly, the transfer points are moved to the top of the loop process, if possible. Then the transfer points are copied from the top of the loop process to a point that post dominates the top of the loop process and the transfer points. (Spec. 5: 5-7; App. Br. 6: 9-21.) A recalculation code is generated and used to subsequently perform the transfer the execution program between the interpreter process and the optimized loop process via the transfer points. (Spec. 5: 8-11.)

Illustrative Claim

Independent claim 1 further illustrates the invention. It reads as follows:

> 1. A program execution method for transferring, from an interpreter process to a loop process of a compiled code process, a

method that is currently being executed for code that includes one or more transfer points at which program execution is transferred from the interpreter process to said loop process of the compiled code process, comprising the steps of:

optimizing the loop process, said optimizing step including the steps of:

moving said one or more transfer points to the top of said loop process if they can be moved there without a problem occurring;

copying code from the top of the loop process to a point that post-dominates said top of said loop process and said one or more transfer points to a location immediately preceding said loop process if said transfer points are located inside said loop process;

storing information for generating recalculation code for one or more specific transfer points when privatization, common sub-expression elimination, and moving of code that are performed pass beyond said specific transfer points; and

performing a recalculation during a transfer process; and

transferring execution from the interpreter process to the optimized loop process via one of said transfer points.

Prior Art Relied Upon

The Examiner relies on the following prior art as evidence of unpatentability:

Koblenz	5,530,866	Jun. 25, 1996
Bak	6,513,156 B2	Jan. 28, 2003

Alfred V. Aho et al., *Compilers: Principles, Techniques, and Tools,* 585-722 (Addison-Wesley, 1986) (hereinafter "Aho").

David F. Bacon et al., *Compiler: Transformations for High-Performance Computing,* ACM Computing Surveys, Vol. 26, No. 4, at 345-420 (1994) (hereinafter "Bacon").

Rejection on Appeal

The Examiner rejects the claims on appeal as follows:

1. Claims 1 through 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Aho, Bak, Bacon, and Koblenz.

2. Claims 5, 6, 8, and 9 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Aho, Bak, and Koblenz.

3. Claims 7 and 10 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Aho and Bak.

Appellants' Contentions

Appellants argue that the proffered combination does not render independent claim 1 unpatentable. (App. Br. 9-13.) Firstly, Appellants argue that Koblenz does not teach moving a transfer point to the top of a

loop process if it can be moved without a problem. (*Id.* at 9.) While the Koblenz reference discloses that, in constructing a tile tree, one can combine all basic blocks of a loop with an outside basic block into a single "summary loop top", it does not convert an irreducible loop into a reducible one. (*Id.* at 10-11.) Secondly, Appellants argue that Aho does not teach copying code from the top of a loop process to a point that post dominates the top of the loop process and a transfer point to a location immediately preceding the loop process if the transfer point is located inside the loop process. (*Id.* at 9.) While the Aho reference discloses copying of a node in a flow graph formed by a plurality of nodes, it does not teach an entry point, a transfer point, and the relationship between them in the graph. (*Id.* at 12-13.)

Examiner's Findings/Conclusions

The Examiner finds that Koblenz' disclosure of a "summary loop top" teaches moving transfer points from the corpus of a flow graph to the top thereof to thereby optimize the flow graph by transforming an irreducible loop into a reducible one. (Ans. 11.) Further, the Examiner finds that Aho's disclosure of copying a node teaches the copying step, as recited in independent claim 1. (*Id.* at 12.)

II. ISSUE

Have Appellants shown that the Examiner erred in finding that:

1) Koblenz teaches moving a transfer point to the top of a loop process if it can be moved without a problem; and

2) Aho teaches copying code from the top of a loop process to a point that post dominates the top of the loop process and a transfer point to a location immediately preceding the loop process if the transfer point is located inside the loop process, as recited in independent claim 1.

III. FINDINGS OF FACT

The following findings of fact (FF) are supported by a preponderance of the evidence.

Koblenz

1a. Koblenz discloses that a loop structure may be used in constructing a tile tree (Col. 7, ll. 63-65.)

1b. A loop top consists of a single basic block with incoming back edges that dominates every basic block in its loop. (Col. 8, ll. 20-23.)

1c. Although an irreducible loop does not contain a loop top, basic blocks in an irreducible loop that are reached by a forward control flow edge from a basic block outside the loop can be combined into a single summary loop top in constructing the tile tree. (Col. 8, ll. 23-27.)

Aho

2. Aho discloses a code optimization scheme in which a single node flow graph is reducible. (Sec. 10.9 at 666, 2nd paragraph "Interval Graphs".)

3. As shown in Figure 10.49, Aho discloses using a node splitting approach followed by an interval partitioning scheme to thereby simplify an irreducible multi-node flow graph into a reducible single node flow graph. Particularly, the node splitting scheme involves splitting or copying a node in the flow graph before it is simplified. (Sec. 10.9 at 666-668.)

4. As depicted in Figure 10.18, Aho discloses using headers for linking different nodes within a single loop. (Sec. 10.4 at 606-607.)

IV. PRINCIPLES OF LAW

Obviousness

Appellants have the burden on appeal to the Board to demonstrate error in the Examiner's position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) ("On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of *prima facie* obviousness or by rebutting the *prima facie* case with evidence of secondary indicia of nonobviousness.") (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)).

> Section 103 forbids issuance of a patent when "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at

> the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains."

KSR Int'l Co. v. Teleflex Inc., 127 S. Ct. 1727, 1734 (2007).

In *KSR*, the Supreme Court emphasized "the need for caution in granting a patent based on the combination of elements found in the prior art," and discussed circumstances in which a patent might be determined to be obvious. *Id.* at 1739 (citing *Graham v. John Deere Co.*, 383 U.S. 1, 12 (1966)). The Court reaffirmed principles based on its precedent that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *Id.* The operative question in this "functional approach" is thus "whether the improvement is more than the predictable use of prior art elements according to their established functions." *Id.* at 1740.

The Federal Circuit recently recognized that "[a]n obviousness determination is not the result of a rigid formula disassociated from the consideration of the facts of a case. Indeed, the common sense of those skilled in the art demonstrates why some combinations would have been obvious where others would not." *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1161 (Fed. Cir. 2007) (citing *KSR*, 127 S. Ct. at 1739). The Federal Circuit relied in part on the fact that Leapfrog had presented no evidence that the inclusion of a reader in the combined device was "uniquely challenging or difficult for one of ordinary skill in the art" or "represented an

unobvious step over the prior art." *Id.* at 1162 (citing *KSR*, 127 S. Ct. at 1741).

V. ANALYSIS

Independent claim 1 recites in relevant part (1) moving a *transfer point* to the top of a loop process if it can be moved without a problem occurring; and (2) copying code from the top of a loop process to a point that post dominates the top of the loop process and a transfer point to a location immediately preceding the loop process if the *transfer point* is located inside the loop process.

We first consider the scope and meaning of the term "transfer point," which must be given its broadest reasonable interpretation consistent with Appellants' disclosure, as explained in *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997):

> [T]he PTO applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification.

See also In re Zletz, 893 F.2d 319, 321 (Fed. Cir. 1989) (stating that "claims must be interpreted as broadly as their terms reasonably allow."). Appellants' Specification states:

> A point that is to be transferred from the interpreter, and a point that will be transferred as a result of examining a method of a program and that will, as a result of the transfer, increase the processing speed are detected, and are defined as transfer points.

(Spec. 8: 1-3.)

Further, Appellants clarify the above definition of transfer point as follows:

> Example (a) shows on the left a loop (nodes 1-3) having an entry point (node 1) at its top as well as *a transfer point (node 2) along its length to which control may pass other than through the entry point (i.e. from another routine)*.

(App. Br. 6) (Emphasis added).

Our reviewing court further states, "the 'ordinary meaning' of a claim term is its meaning to the ordinary artisan after reading the entire patent." *Phillips v. AWH Corp.,* 415 F.3d 1303, 1321 (Fed. Cir. 2005). The Examiner found that Aho teaches transfer points as headers in a loop. (Answer 10.)

Based upon the record before us, we construe a transfer point as a point (other than an entry point) in a loop through which control passes to or from another routine or loop.

As set forth in the Findings of Facts section, Aho discloses an optimization scheme whereby a node splitting technique followed by an interval partitioning approach are used to transform an irreducible multi-node loop structure into a reducible single one. (FF. 2-3.) Further, Aho

discloses using two headers for linking different nodes within a single loop. We find that while the Aho reference teaches a loop optimization scheme to simplify an irreducible loop structure into a reducible one, it does not teach using transfer points for performing such simplification. Particularly, Aho's teaching of headers does not comport with the definition of transfer points that we adopted hereinabove. We find that Aho's headers are limited to be used within a single loop. In our view the disclosed headers are not intended to pass or receive control to another loop or routine, as a transfer point would.

Additionally, as detailed in the Findings of Facts section, Koblenz discloses combining the basic blocks of an irreducible loop structure with another basic block outside the loop structure by using a forward control flow edge from the outside basic block to thereby form a single summary loop atop a constructed tile tree. (FF. 1a-1c.) While the Koblenz reference teaches using a forward control flow edge to combine the basic blocks of a loop with another basic block outside the loop structure, it is silent as to whether the outside basic block is part of another loop structure or another routine. It is further silent as to whether control is being passed to or from the outside basic block to the basic blocks in the loop structure. Consequently, we find that the forward control flow edge does not comport with the definition of the transfer point as defined hereinabove. We therefore agree with Appellants that neither Koblenz nor Aho teaches using one or more transfer points to optimize the loop process. It follows that

Appellants have shown that the Examiner erred in finding that the combination of Aho, Bak, Bacon, and Koblenz renders independent claim 1 unpatentable.

Because independent claims 4, 5, 7, 8, and 10 also recite optimizing a loop process using one or more transfer points, we find that Appellants have also shown error in the Examiner's rejections of these claims, as well as the claims depending therefrom.

VI. CONCLUSION OF LAW

Appellants have shown that the Examiner erred in rejecting claims 1 through 10.

VII. DECISION

We reverse the Examiner's decision to reject claims 1 through 10.

REVERSED

msc

WILLIAM A KINNAMAN JR.
IBM CORPORATION - MS P386
2455 SOUTH ROAD
POUGHKEEPSIE NY 12601